
INTERACTIVE JAVA TUTORIAL FOR E&B-LEARNING FLEURY'S ALGORITHM¹

M Gloria Sánchez-Torrubia, Carmen Torres-Blanc, Víctor Giménez-Martínez

Abstract: *Graphs and graph algorithms, play a very important role in the Informatics Engineer's background. In this paper, a tool implemented as a Java applet and designed as an auxiliary instrument for both classroom teaching and individual practicing of Fleury's algorithms is presented. This tool, included within a set of active learning tutorials, has been designed as educational complement for Discrete Mathematics first course students. Its characteristics of visualization, simplicity and interactivity, make this tutorial a great value pedagogical instrument, to be used both by the professor in the classroom, as well as by the students in their individual learning.*

Keywords: *e&bLearning, education, active learning, interactive Java applets, discrete mathematics learning, visualization.*

ACM Classification Keywords: *K.3.1 [Computers and Education]: Computer Uses in Education – computer-assisted instruction (CAI), distance learning. K.3.2 [Computers and Education]: Computer and Information Science Education – computer science education, self-assessment. G.2.2 [Discrete Mathematics]: Graph Theory – graph algorithms, path and circuit problems.*

1. Introduction

At the present time, new technologies are producing an important change in our society. Education cannot keep out of this development and technological advances should generate a substantial change in our education systems. Thus, Web based learning technologies play a very important role in the modern education process and extensive research shows increasing use of computers in education. Some investigations have shown that computer assisted instruction has been more effective than traditional methodologies [Waldock 2002]. However, comprehensive research is required to determine the best methodology to be applied to the design and development of computer-assisted training, as well as the efficiency of the teaching/learning processes based on this particular method of instruction [Hundhausen 2002].

Furthermore, the big amount of stimuli students are receiving from their environment cause that, in contrast, traditional classes seem less appealing. In this context, it is evident that introducing new incentives in the teaching-learning process is becoming compulsory. It is well known that computer assisted education provides new exciting tools for the development and usage of teaching and learning resources. That is why blended courses, taking advantage of computational aids, are turning into a must especially in computer and engineering education. This way, the power and effectiveness of face to face teaching are enhanced with the flexibility and technical capabilities of eLearning, turning out the students in the major figures of their learning progression.

The background the students bring, both from their previous learning processes and from the social atmosphere, is also transforming their reasoning abilities. Authors' teaching experience shows that Informatics Engineering students have significantly increased their algorithmic reasoning capability, both in comprehension and design,

¹ This work has been partially supported by UPM under project No. **IE06 1010-092**

tallying with an important decrease of their formal and algebraic reasoning ability. Thus, fostering graphs' algorithmic approach should be decisive on enhancing students' logical potentials.

2. Impact of Java Interactive tools on teaching and learning

Graphs and graph based algorithms are mathematical structures which will take an important role in Informatics Engineer's background. The Polytechnic University of Madrid has included, in its Informatics Engineering first course program, a graph theory introduction as a main section of the Discrete Mathematics subject. Because of it, the authors were stimulated for designing and developing several tutorials, whose purpose is to enhance the comprehension and learning of this matter to first year course students.

A comparison study between a control section and a computer-based studio section of calculus-based Introductory Physics performed in Arcadia University (Canada) [Retson 2000] showed that the two groups were statistically different, with the studio class outperforming the traditional lecture class in all cases. Collectively, their results clearly proved that the studio model was significantly more effective than the lecture model.

Recent approaches in the aim have shown that visualization aided interactive tools provide a very positive aid to the learning task [Naps 2003a, Sánchez-Torrubia 2006b, Torres-Blanc 2006]. Furthermore, mathematical concepts and algorithmic procedures as well, are often difficult to be explained in the classroom. That is why visualization characteristics of these tools play an important role, increasing insight –an image is worth thousand words–, engagement, memorization and satisfaction for the students. Additionally, the interest and motivation among pupils is increased when the teacher uses those tools.

Visualization is a very important feature when learning graph algorithms, but the main characteristic of the Fleury's tutorial presented here is its full interactivity. This aspect is a good aid for teachers, as user's examples may be exposed and the application gives time for explaining difficult points. However, the major objective when pursuing the full interactivity has been the student's learning task. This way, as will be described in section 3.3, the user must execute the algorithm while the tool evaluates and corrects the process.

2.1. Assisting teacher's task

Graph algorithms are dynamic structures difficult to show in static tools such as blackboards or slides. Therefore, the feasibility of applying dynamic visual tools to represent those concepts shows an enormous potential for achieving high didactical goals [Naps 2003b, Sánchez-Torrubia 2006a, Sánchez-Torrubia 2001].

Many educators think that visual tools enhance their lectures and produce significant increase in student's comprehension, but such tools are of little effectiveness when students are not actively engaged in the learning process [Naps 2003a, Röbling 2000, Sánchez-Torrubia 2006b] or teachers cannot present their examples.

According to the conclusions of the Working Group "Improving the Educational Impact on Algorithm Visualization" [Naps 2003a], when a group of teachers were asked about the benefits they had experienced from using visualization, they got, among others, the following results: 90% declared that teaching experience was more enjoyable, 86% noticed an improved level of student participation, 83% had anecdotal evidence that the class was more exciting for students, 76% thought that visualization provided a powerful basis for discussing conceptual foundations of algorithm, and 52% got objective evidence of improved student learning.

2.2. Assisting student's learning process

Active learning Java tools do enhance teacher's lectures, but their best feature lies on the possibilities they show when used by the students themselves.

As it was demonstrated in *Deakin University* (Australia) [Street 1998] students who used interactive tools learned 60% faster, and after 30 days the knowledge kept was from 25% to 50% higher than those who did not use them.

Graphical and dynamical visualizations are more appealing for learners than exercises or text books, but, if the students are not required to give some kind of answers or predict what is happening next, they might adopt a passive attitude that is not beneficial at all and might even be prejudicial to their training. It has been verified that learners spend much more study time when visualization is involved, but those who are actively engaged have consistently outperformed the other ones who passively viewed them [Hundhausen 2002]. That is why, during the execution, the program should in some way ask the user which must be the next step to be done, not just show it. Here lays the main difference between an active learning tool, as the one presented in this paper, and a simple demonstrative visualization tool.

3. Fleury's algorithm interactive tutorial

3.1. Fleury's algorithm

Fleury's algorithm is designed for finding an Euler Path in an undirected graph. The graph has an Euler path if it is possible to start at a vertex and move along the graph so as to pass along each edge without going over any of them more than once.

This problem has its origin in Leonhard Euler's work directed towards the geometry of position which consists of a single paper [Euler 1736], now considered to be the starting point of modern graph theory. In it, Euler undertakes a mathematical formulation of the famous Königsberg Bridge Problem: is it possible to plan a stroll through the town of Königsberg which crosses each of the town's seven bridges once and only once? In this paper, Euler's main result is, in a modern statement, the well known theorem: A finite graph G contains an Euler path if and only if G is connected and contains at most two vertices of odd degree. He also sketches a procedure for finding the path consisting on creating a simple circuit, eliminating the used edges, finding new circuits in the remaining graph and joining the new circuits in the proper vertexes. This procedure, very intuitive in theorem's formal demonstration, is not algorithmically effective.

The algorithm presented here is credited to a mostly unknown French mathematician named Fleury [Fleury 1883]. It starts with a vertex of odd degree —if the graph has none, then start with any vertex—. At each step it moves across an edge which is included in a cycle, unless there is no choice, and then we delete that edge.

3.2. Algorithm's description

The basic idea is that when drawing an Euler circuit, all passed edges cannot be used again. So, at any moment in drawing, with all passed edges deleted, the remaining edges must be in one connected component.

Fleury's algorithm pseudocode might be described as follows:

Input: A connected graph $G = (V, E)$ with, at most, two odd grade vertexes, where V is the set of vertexes and E is the set of edges.

Output: A list $P = v_0e_1v_1e_2, \dots, e_i v_i e_{i+1}, \dots, e_m v_m$, representing the path which includes each edge in E exactly once.

Procedure:

If an odd grade vertex v_0 exists, *then*

$P = v_0$ where v_0 is any odd grade vertex of V

else

$P = v_0$, where v_0 is any vertex of V

End if

$\varepsilon = |E|$; $v = v_0$; $i = 1$

```

While  $i \leq \varepsilon$  do:
     $E_v = \{e \in V \mid e \text{ is adjacent to } v\}$ 
     $e_i = \text{any edge of } E_v$ 
    While  $e_i$  is a bridge (an edge which is not included in a cycle) and  $|E_v| > 1$  do
         $E_v = E_v - \{e_i\}$ 
         $e_i = (v, w)$ , any edge of  $E_v$ 
    end while
     $v_i = w_i$    $P = P \cup e_i$    $E = E - \{e_i\}$ 
     $v = v_i$    $i = i + 1$ 
end while
return  $P$ 

```

For deciding whether e_i is a bridge or not, a modification of Depth-First Search algorithm [Sánchez-Torrubia 2006c] should be used.

3.3. Tool's description¹

The tutorial contains definitions, examples and theoretical explanations of the basic concepts of graph theory needed for understanding Fleury's algorithm, its pseudocode and an interactive applet implementing it.

The main feature of this tool is its full interactivity. It means that the user can introduce the graph and execute the algorithm while the application evaluates the inputs provided by the student. In other words: in real time, the applet will only evaluate the input introduced by the user. If it is right, the application will implement the order, and then it will remain in a stand by mode, waiting for a new one. If the input is not right, an error message will appear on the message window, indicating to the user, what the error is and waiting for the right one. Once the algorithm has been completed, a successful '*end of algorithm*' message will be displayed.

The application has a message window, used for displaying error messages and providing next step hints.

3.3.1. Introducing the graph

In this applet, graph nodes are introduced by left clicking on the drawing area. The edges are drawn by clicking successively two nodes. At any execution moment, the user may interrupt the process in order to add new nodes and/or edges, and restart it once again.

3.3.2. Executing the algorithm

Once the graph has been introduced by the student and the algorithm is running, the application checks whether an eulerian path exists. If the graph is not connected or there are more than two odd degree vertexes, an error message is displayed indicating the corresponding fault. If there is an eulerian path, the applet asks the user to click on the initial vertex. Subsequently, in each step, the user should click the next vertex to be visited and, if the entry is correct, the application will change the selected edge's color and add a number indicating the path sequence (Figure 1).

¹ DMA: Algoritmo de Fleury. <http://www.dma.fi.upm.es/java/matematicadiscreta/fleury/>

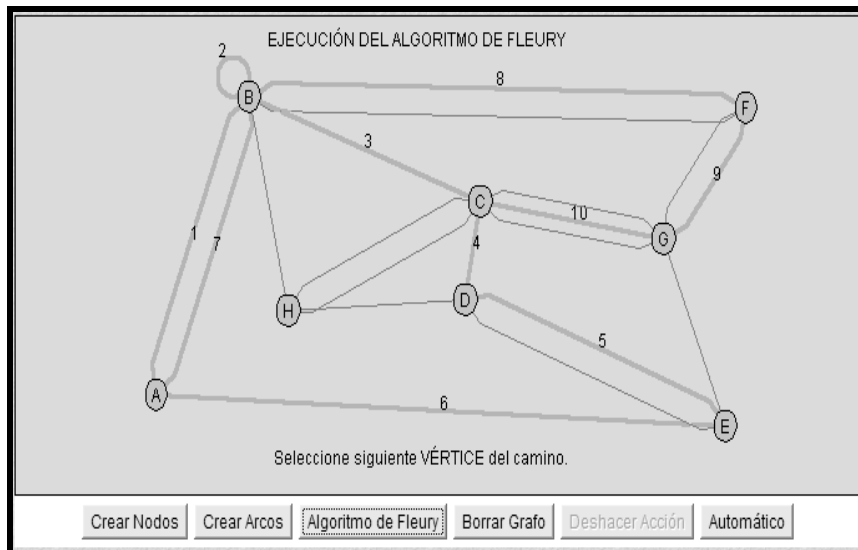


Figure 1: Fleury's applet in the process

4. Conclusion

The application has been designed with the main purpose of supporting active learning. Being interactive, easy to use, intuitive and visual are the characteristics kept in mind during design and implementation phases. Actually, these qualities have demonstrated to help achieving increased engagement for the students when attending a teacher's lecture, as well as when working by themselves.

In our experience, java interactive tutorials are very good aids for learning graph algorithms as they improve comprehension, engagement, memorization and satisfaction for the students, so as the interest and motivation among pupils when the teacher makes use of them.

As mentioned above, Informatics Engineering students have significantly increased their algorithmic reasoning capability, both in comprehension and design. That's why this interactive java applet together with some others also available or in development process in Applied Mathematics Department web site¹ will surely help on teaching and learning graphs' algorithms, a subject that will be decisive on enhancing students' logical potentials.

Bibliography

- [Euler 1736] Euler, L. *Solutio problematis ad geometriam situs pertinentis*. *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8, 128-140 (1736). Based on a talk presented to the Academy on 26 August 1735.
- [Fleury 1883] Fleury. *Deux problemes de geometrie de situation*. *Journal de mathematiques elementaires* 1883, 257-261.
- [Hundhausen 2002] Hundhausen, C. D., Douglas, S. A. and Stasko, J. T. *A Meta-Study of Algorithm Visualization Effectiveness*. *Journal of Visual Languages and Computing*, 13, 3, Elsevier, 259-290 2002.
- [Naps 2003a] Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., Velazquez-Iturbide, J. A.: *Exploring the Role of Visualization and Engagement in Computer Science Education*. *Inroads - Paving the Way Towards Excellence in Computing Education*. pp. 131-152, ACM Press, 2003.

¹ <http://www.dma.fi.upm.es>

- [Naps 2003b] Naps, T. L., Lucas, J., Röbling, G. VisualGraph – A Graph Class Designed for Both Undergraduate Students and Educators. Proc. of the 34th ACM SIGCSE Technical Symposium on Computer Science Education (2003) pp. 176-171.
- [Retson 2000] Retson, D., Williams, P. J., Simmons, S.: The Effectiveness of Computer-Based Studio Teaching of Physics. <http://aitt.acadiau.ca/research/science/PhysicsEdStudy5.PDF> (last visited January 2007).
- [Röbling 2000] Röbling, G., Freisleben, B.: Experiences in Using Animations in Introductory Computer Science Lectures. Proc. of the 31th ACM SIGCSE Technical Symposium on Computer Science Education (2000) pp.134-138.
- [Sánchez-Torrubia 2006a] Sánchez-Torrubia M. G., Sastre-Rosa, M. A., Giménez-Martínez, V., Escribano-Iglesias C. Pedagogical impact of interactive tutorials in visualization and learning of mathematical concepts in computer science curricula. Proceedings of the Conference on Informatics Education in Europe (Montpellier, France, 9–11 November, 2006).
- [Sánchez-Torrubia 2006b] Sánchez-Torrubia M.G, Giménez-Martínez, V. Java Tutorials: a good tool for teaching and learning Graph Algorithms. Proceedings of the 3rd International Conference on the Teaching of Mathematics (ICTM3). (Istanbul, Turkey, 30 June, 5 July 2006).
- [Sánchez-Torrubia 2006c] Sánchez-Torrubia M.G., Gutiérrez-Revenga S. Tutorial interactivo para la enseñanza y el aprendizaje de los algoritmos de búsqueda en anchura y en profundidad. Proceedings of the XII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2006). (Bilbao, Spain, 12–14 July 2006). Thomson, 573-580.
- [Sánchez-Torrubia 2001] Sánchez-Torrubia M.G., Lozano-Terrazas V. Algoritmo de Dijkstra: Un tutorial interactivo. Proceedings of the VII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2001). (Palma de Mallorca, Spain, 16–18 July, 2001). J. Miró, 254-258.
- [Street 1998] Street, S., Goodman, A. Some experimental Evidence on the Educational Value of Interactive Java Applets in Web-based Tutorials. 3rd Australasian Conference on Computer Science Education. Association for Computing Machinery (1998) pp. 94-100.
- [Torres-Blanc 2006] Torres-Blanc, C., Sánchez-Torrubia, M.G., Aguilar, J., Castiñeira-Holgado, E. Analyzing and Applying Computer Algebraic Systems to Engineering Education: an Interactive Digital Control course WSEAS Transactions on Advances in Engineering Education Vol. 3 N° 2 (November 2006). WSEAS Press, 977-983.
- [Waldock 2002] Waldock, J., Gretton, H., Challis, N. Using the web to enhance student learning. Proceedings of the 2nd International Conference on the Teaching of Mathematics (Crete, Greece, July 1-6, 2002).

Authors' Information

M Gloria Sánchez-Torrubia – Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s.n., 28660 Boadilla del Monte, Madrid, Spain; e-mail: gsanchez@fi.upm.es

Carmen Torres-Blanc – Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s.n., 28660 Boadilla del Monte, Madrid, Spain; e-mail: ctorres@fi.upm.es

Victor Giménez-Martínez – Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s.n., 28660 Boadilla del Monte, Madrid, Spain; e-mail: vgimenez@fi.upm.es